Expert Decision Support Based On the Generalized Network Model: An Application To Transportation Planning

Spyros Tzafestas and George Kapsiotis

Intelligent Robotics and Control
Computer Science Division
National Technical University of Athens
Zographou 15773, Athens
GREECE

T. Pimenides

Systems and Control Division
Department of Electrical Engineering
University of Patras
Patras 26500
GREECE

Abstract: In this paper an expert (knowledge-based) decision support system (KBDSS) architecture is presented which is then specialized to the generalized network (GN) domain. The resulting KBDSS developed by the authors and called GENETEXP is actually useful for decision making in real problems that may be put in the GN form and need additional higher level expertise. Some results in the optimized transportation planning area are included to illustrate the functioning and usefulness of the KBDSS.

1. Introduction

Techniques from the disciplines of operational research (OR), decision analysis (DA) and artificial intelligence (AT) have been extensively investigated over the years and recently used in the development of computerized decision aids. Such systems alone have been found to have significant limitations in handling practical problems. Conventional decision support systems (DSS) are usually built around prescriptive models which however are very rigid and do not have the required flexibility. On the other hand AI/knowledge-based systems (KBS) tools show a lack of established techniques for problem structuring and knowledge acquisition.The AI and DSS fields currently draw attention of many workers to the end of integrating and unifying such techniques and tools. The ultimate goal is to beneficially employ the problem structuring aspects of OR/DSS and the incrementally modifiable properties found in AI/KBS tools. Several conceptual, methodological and implementation issues along this path of DSS/KBS integration can be found in [1-6].

In this paper a general architecture of integrated KBS-DSS (or KBDSS) is presented and its main properties and requirements are discussed. Then a new **generalized network** (GN) - based software tool (the GENETEXP) developed by the authors is briefly described. GENETEXP is actually an integrated knowledge-based software environment for decision- making in practical problems that are dominated by a GN model. The tool is under further development particularly in the direction of enhancing the capabilities of its knowledge-based component. For completeness a brief overview of the major GN algorithms developed over the years (Jewell's algorithm, a dual algorithm, relaxation algorithm) and incorporated in our KBDSS is provided.

2. A General Architecture for Knowledge Based Decision Support Systems

According to Elam et al [7] a conventional DSS consists of four basic components, namely: **control subsystem**, **data subsystem**, **model subsystem** and **report subsystem** (Figure 1). The functioning of each one of them is as follows:

CONTROL SUBSYSTEM : This subsystem connects the data, model and report subsystems, and provides the primary user interface for system operation. The user sends via this interface appropriate instructions to be carried out by the DSS in order to perform a particular decision analysis. The control subsystem should be able to consult each of the other subsystems, as necessary, to meet the user's request(s) in addition to satisfying the job step and file manipulation requirements of the host computers. Actually, the control subsystem frees the decision maker from the requirement of mastering any operating procedures of his/her computer(s). He (she) has only to understand the problem situation and analysis procedures.

The design of the user interface is of primary importance. It should consist of a set of high level, English-like language statements which can be used to produce the instructions about carrying out a certain analysis in accordance with the manager`s conceptualization of the problem situation.

DATA SUBSYSTEM : This subsystem organizes and maintains all the data needed for the implementation and operation of the KB decision support system (KBDSS). It should involve

- data definition (specification and organization of the data items)
- data query (to retrieve data items from the database)
- data manipulation (to update or review data items for an alternative scenario analysis)

A database management system can be used to efficiently perform the above tasks.

MODEL SUBSYSTEM : This subsystem enables the KBDSS to consider all relevant processes/activities in an integrated way. This is done by quantifying the combined impact over a planning horizon of resource restrictions, costs, management policies, and market conditions on alternative logistical tactics, etc. The model subsystem involves two particular functions:

- model definition and generation
- model solution, to determine the recommended supply, movement and storage plan.

The model should exhibit the appropriate degree of realism (with regard to the planning objectives) and also be understandable.

REPORT SUBSYSTEM : This is an important link in the communication process between the model subsystem and the users of the KBDSS. The report subsystem accesses the basic data and model solutions, which became a part of the database. From these data, it generates the set of reports specified by the user. Report facility options include printed copy, CRT copy, printed graphics, and CRT graphics.

The DSS architecture described above can now be generalized by adding a **knowledge-based** subsystem as shown in Figure 1. This subsystem employs AI/expert system techniques and enhances management support by incorporating into the system the knowledge of analysis, decision- makers, and experts in the field.

It involves the following principal functions:

- Input data analysis (to analyze, verify and modify the data which support the system)
- Parameter changes (to automate tactical what-if analysis by resetting certain parameters of the model or to set parameters based on logical conditions). Parametric changes may be used to study the sensitivity of the optimal solution to changes in the data values.
- Post- optimality analysis (to extract key information from the model solution and from several reports in order to facilitate evaluation of the model proposal).

The overall knowledge-based decision support system (KBDSS) architecture of Figure 1 can also be cast in the structure of the KBDSS proposed in [8] where three principal components are considered, namely a **language system** (LS), a **knowledge system** (KS) and a **problem processing system** (PPS) (Figure 2). The language system is the total of the linguistic facilities made available to the decision maker by the DSS, and plays the role of a vehicle that allows that the decision- maker or user conveys information to the DSS. The user put his (her) problem to the DSS via the LS, and in addition the LS is a two -way avenue through which the decision maker and the PPS can interact. In the architecture shown in Figure1 the LS consists of the control subsystem, data subsystem and (partly) the report subsystem. The syntactic and semantic rules of a LS determine the allowable problem statements that can actually be posed to the DSS. All languages from fully procedural to fully non procedural (i.e. AI) languages can possibly be used according to the particular nature of the problem(s) domain the KBDSS is to be used in. A natural language is ideal from the user`s point of view but a full natural language (such as English) is difficult to parse because of its context-sensitive characteristics. Several research studies are currently conducted in this important area for the next generation KBDSS[9-11].

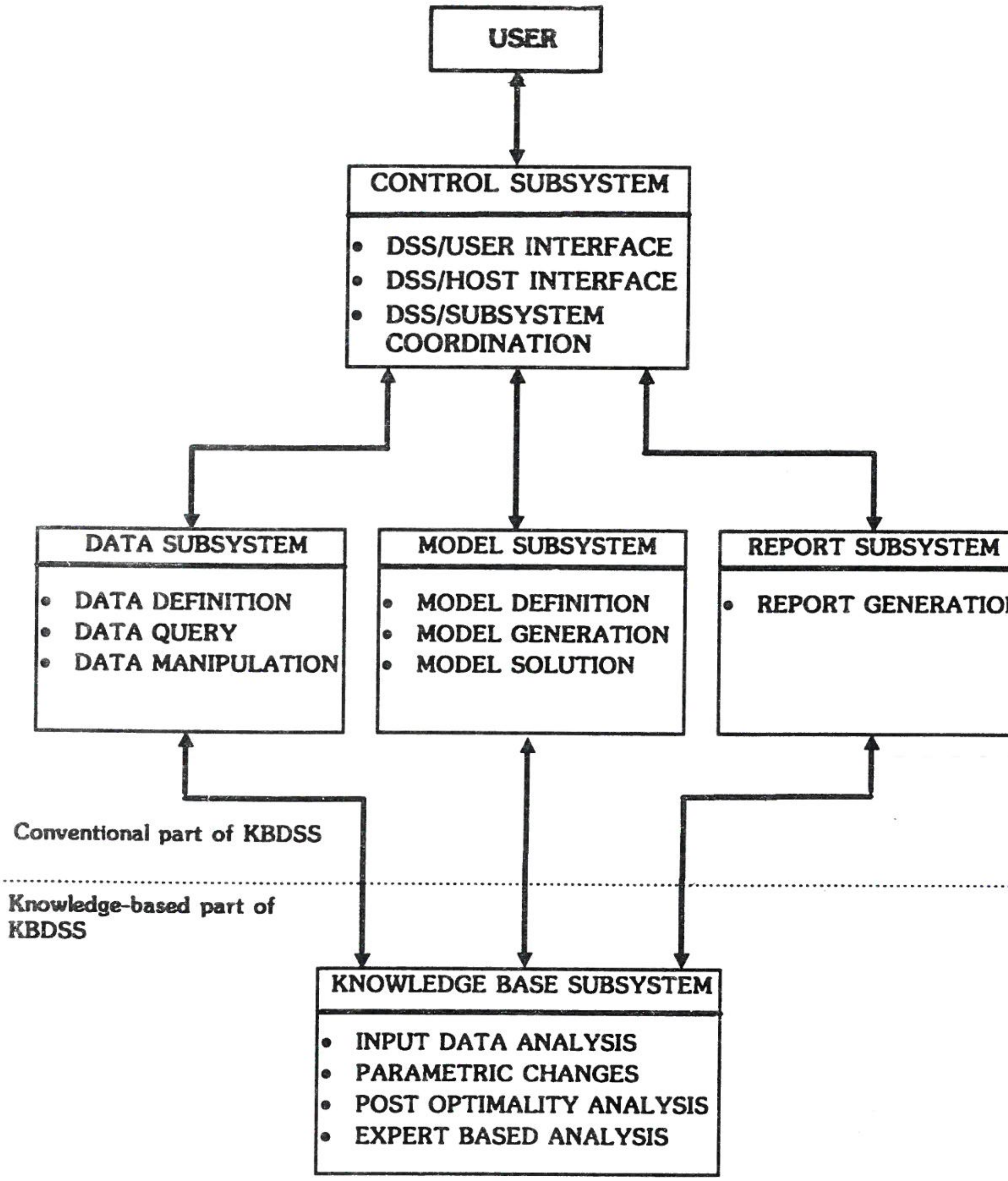


Figure 1. General Architecture of Knowledge-based Decision Support System

The **knowledge system** (KS) provides all required facilities for the representation and organization of knowledge. Several knowledge representation methods can be used here (predicate calculus, AI production systems, rules, objects, etc.). Very frequently it is desirable to use in an integrated way more than one knowledge representation method combined with (relational, hierarchical or network type)database systems.

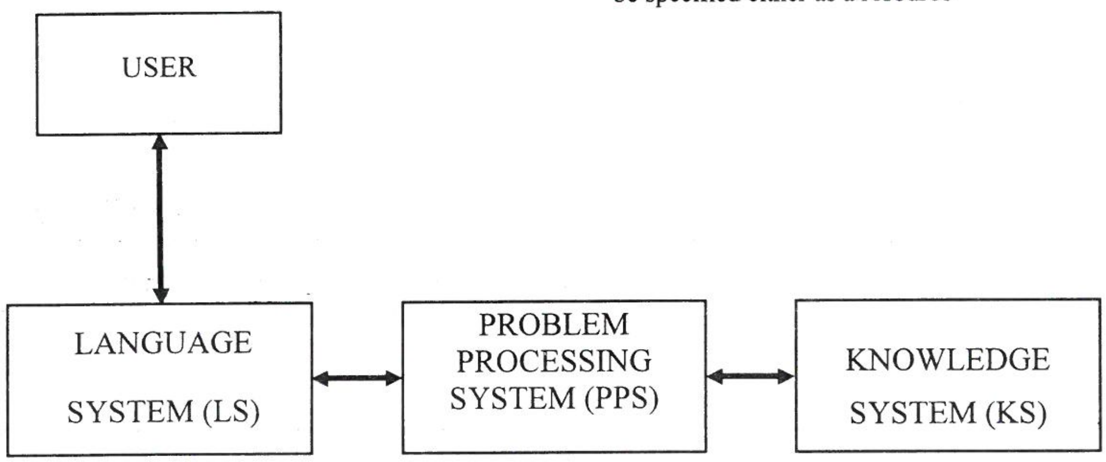


Figure2. An Alternative Structure of KBDSS

The **problem processing system** (PPS) (i.e. the **model subsystem** in Figure 1) is the formal specification of the DSS, behaviour patterns, and must be capable of recognizing problems by transforming problem statements into suitable executable plans of action which, when executed, yield a solution to the problem. If a model is specified or selected by the user, then there is no need for the PPS to recognize the modelling problem. But if the PPS has itself to select or formulate the model, a highly sophisticated problem recognition facility must be incorporated into the PPS. An additional important PPS ability is that of analysis i.e. the process of interfacing models with data in order to generate assertions. Some useful ways of implementing a PPS can be found in [12-14].

3. The GENETEXP Tool

GENETEXP is a tool for analyzing GN problems within generalized decision making environment. With this package, any problem that has the mathematical formulation (1) can be modelled and analyzed. Actually, the problems to be treated with GENETEXP are of a broader scope than the one stated in (1), as it can deal with boundary condition constraints of inequality type, i.e. the nonzero supply for a given node can be specified either as a resource to be compulsorily conveyed through network, or as the available quantity of the resources considered. In the same way, the value of a demand can be considered either as a specified requirement to be met, or as a minimum of the required quantity. The architecture of GENETEXP is as shown in Figure 1. The user controls the execution of the program interactively, by selecting among the "active" options of a hierarchically organized menu. The structure of the menu, accompanied by a brief description of each of its functions, is given in Figure 3. The routines implementing the mathematical algorithms necessary for the solution of the model are aggregated into the Model Subsystem. The role of the Data and Report subsystems is self-explanatory. Care has been taken so that the program is user-friendly and robust to any user`s mistypings.

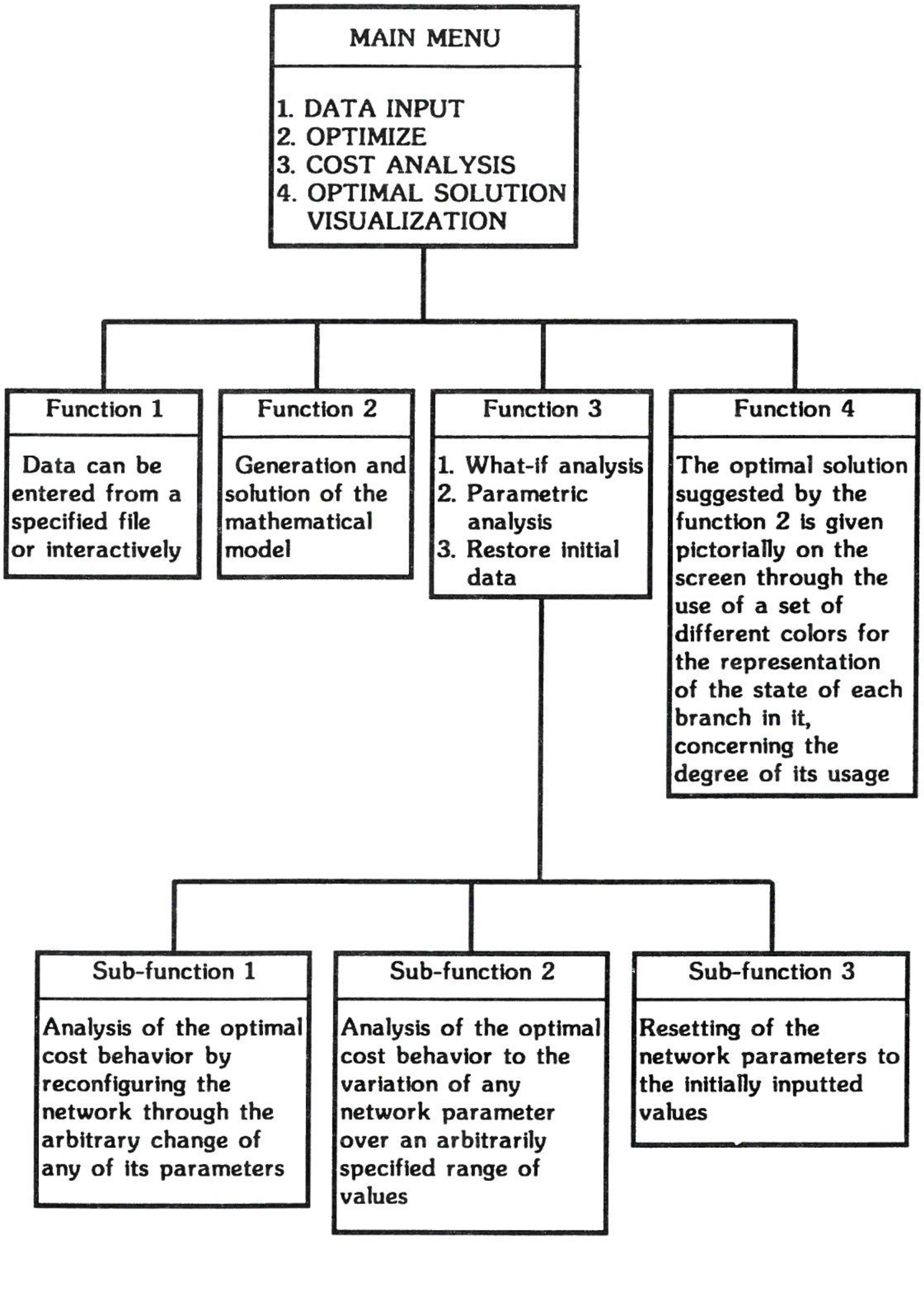


Figure 3 Mathematics - based Genet Functions

GENETEXP is the outcome of the integration of GENET OPTIMIZER (a previous pure numeric GN-DSS package [21]) with an expert system shell developed at NTVA and used extensively for engine fault diagnosis (ENGEXP) and medical therapeutic treatment (BIOEXP) [22-23]. The package was developed in PASCAL under DOS on an IBM-PC compatible with sufficient memory to support the model's data and knowledge structure. A number of experiential rules have been accommodated to the knowledge base for heuristic (expert based) post optimization analysis. Work is under development by the authors to incorporate into the knowledge base subsystem fully heuristic models accompanied by user-decision maker preferential rules. This is done in co-operation with European-level supply-distribution chain. The system will then be capable of evaluating the solutions and use the one that satisfies not only numerical but also other experiential preferential requirements.

4. Brief Review of GN Algorithms

4.1 Statement of the GN Optimal Flow Problem

A **network** is defined by a set V of **vertices/nodes** i and a set B of **directed branches** (i,j) which is a subset of V×V. With every branch we associate a non negative flow X_{ij} running through it, the **accepted interval** (L_{ij} , M_{ij}) for the flow values, the cost C_{ij} for every unit of flow transferred through the branch, and a **gain/multiplier** K_{ij} which modifies the incoming flow X_{ij} to the outgoing $K_{ij} \cdot X_{ij}$. Nodes are classified as i) **supply nodes**, where positive flow S_i is input into the network, ii) **demand nodes**, where flow D_i leaves that network, and iii) **transshipment nodes**, where branch-flow running into the node balances the branch-flow running out of it.

A pictorial representation of the above notions has been established in order to facilitate network modelling, the **NETFORM** presentation. A typical branch in this representation looks like that of Figure 4. In this Figure, the representation of modal supply and demand is also given.

The mathematical statement of the problem results directly from its verbal expression and the parameter definitions given above. It reduces to the following Linear Programming (LP) problem:

$$min\ c = \sum\nolimits_{(i,j)\in B} C_{ij} X_{ij} \qquad \text{(a)}$$

$$sbj.\ to \sum\nolimits_{j,(i,j)\in B} (X_{ij} - K_{ji} X_{ji}) = S_i - D_i\ ,\ i \in V \qquad \text{(b)} \quad (1)$$

$$L_{ij} \le X_{ij} \le M_{ij}\ ,\ (i,j)\in B \qquad \text{(c)}$$

The A-matrix of this linear problem has the special property that at most only two entries in each column are non-zero; in fact, this remark constitutes the principal criterion for the identification of a GN structure. By scaling and/or complementing to variable's upper bounds the matrix and right-hand-side coefficients, it is always possible to make one of the non-zero entries of each column equal to-1. Taking the corresponding node to be the branch start, the other coefficient is equal to the branch-gain. Usually, the constraint $L_{ij} \le X_{ij}$ in formulation (1) is handled by setting $X'_{ij} = X_{ij} - L_{ij}$ and transforming problem (1) into another one in variables X'_{ij} , $0 \le X'_{ij} \le M_{ij} - L_{ij}$ and with the right-hand-side vector appropriately modified. In what follows, we assume that this transformation has taken place although we use unprimed symbols.

4.2 Solution Algorithms

Since problem (1) is a LP program, it can be treated by the Simplex method in any of its variations. This method is established by *the fundamental theorem of Linear Programming*, and constitutes a systematic searching among the set of basic solutions (i.e. solutions in which the number of variables with a non-zero value is equal to the rank of the A-matrix) for the optimal one (i.e. the one that satisfies (1a)). However, the special structure of the A-matrix gives rise to the following theorem concerning the basis topology:
Theorem: *Any subset B_b of B spanning all vertices V constitutes a basis if and only if it consists of a number of components of connectivity with each component having only*

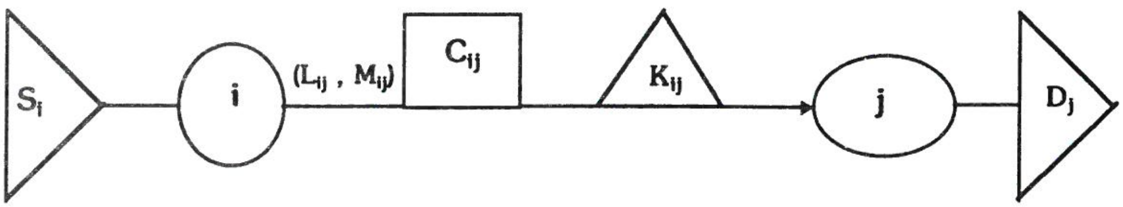


Figure 4. NETFORM Presentation

one loop in it and the loop-gain is different from unity.

The **loop-gain** is related to a direction of transversion and it is defined as the product of gains of the branches transversed in the direct sense divided by the product of gains of the branches transversed in the reverse sense. A loop with gains equal to unity is said to be *degenerate*. The conceptualization of the problem structure and its primary features has led to the development of specialized algorithms for its solution. According to these approaches, data structures are suggested by the GN concepts and the underlying topologies, while computations make explicit use of these concepts and are related directly to their schemes. Even though the developed algorithms are tailored to the GN structure, they depart from the Simplex method and make extensive use of the relevant theory of their validation. Thus, they can be classified as *Primal*, *Dual*, and *Primal-Dual*, in a similar fashion. Primal and Primal-Dual algorithms appear to be the most efficient ones, while it is established that duality and dual methods are most appropriate when it comes to post-optimality aspects. During the past thirty years, a variety of algorithms belonging to the above three categories has been developed. Some of them treat the problem to its general formulation, while some others are adapted for the solution of specific GN configurations. In [15-17] a citation of the algorithms and codes developed until that time (late `70s) is made. In what follows, a Primal-Dual algorithm developed by Jewell in 1962 is firstly presented [18], while in continuation, a dual algorithm developed by the authors [19], is discussed; finally, a more recent approach, which has been developed by Bertsekas and Tseng at MIT and is classified as a *relaxation method*, is outlined [20].

4.3 Jewell`s Algorithm

In relation to the problem (1), which is characterized as the **Primal** problem, the **Dual** problem is defined, in variables U_{ij} , V_i (corresponding to the network branches and nodes) as follows [18]:

$$\max c = \sum_{i\in V}(S_i - D_i)V_i - \sum_{(i,j)\in B} M_{ij}U_{ij}$$

$$\text{sbj. to } V_i - K_{ij}V_j - U_{ij} \le C_{ij} \qquad (2)$$

$$U_{ij} \ge 0$$

$$V_i \text{ unrestricted}$$

To solve the problem by the algorithm considered, it must be transformed into the **canonical form**. The transformation is a two-step process: first, the least-flow constraints are incorporated into the nodal boundary conditions, and in continuation, artificial nodes and branches appropriately capacitated are added, so that there remains only one boundary node (condition). The transformation technique is presented in Figure 5.

The transformed LP is:

$$\min c = \sum_{(i,j)\in B} C_{ij}X_{ij}$$

$$\text{sbj. to } \sum_{j\in V}(X_{ij} - K_{ji}X_{ji}) = \begin{cases} Q, & i = 0 \\ 0, & i \ne 0 \end{cases}$$

(3)

$0 \leq X_{ij} \leq M'_{ij}$

On generalizing the Ford and Fulkerson techniques for pure networks, a **restricted Primal** problem is formulated and solved at every cycle of the algorithm. Specifically, by stipulating non-negative costs, the algorithm is initiated with a zero dual feasible solution and zero network-flow; in this way, all of the constraints in (3) are satisfied except for the balance in the reduction of the non-feasibility of the running network-flow, or in the resolution of the problem's non-feasibility.

The formulation of the restricted Primal is as follows:

$$\max F_0$$

$$\text{sbj. to } \sum_{j \in V} (X_{ij} - K_{ji} X_{ji}) = \begin{cases} F_0 , & i = 0 \\ 0 , & i \neq 0 \end{cases} \qquad (4)$$

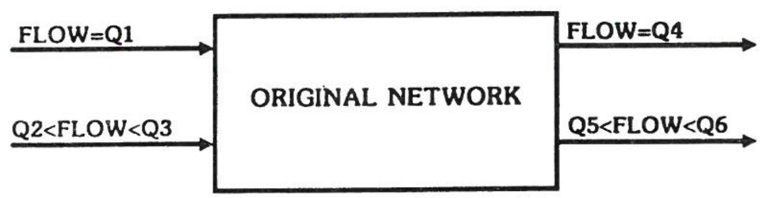
$$0 \leq X_{ij} \leq M'_{ij} \quad \text{for } V_i - K_{ij} V_j = C_{ij}$$

$$0 = X_{ij} \quad \text{for } V_i - K_{ij} V_j < C_{ij}$$

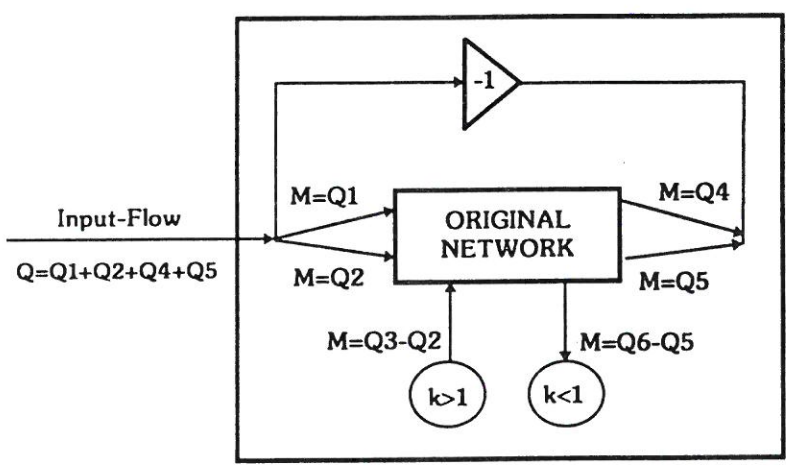
$$X_{ij} = M'_{ij} \quad \text{for } V_i - K_{ij} V_j > C_{ij}$$

It is obvious that (4) constitutes a *maximal-flow* problem for an appropriately specified part of the original network. To solve this problem, the notions of the *generating* and *absorbing* loops are exploited; it is demonstrable that a loop with loop-gain greater than unity can operate as a source of flow for the entire network, while a loop-gain less than unity can operate as a sink. Thus, the maximal-flow routine is reduced to the identification of the absorbing structures of the restricted network, and the determination of the maximal-flow they can absorb, in an incremental mode. This is done through a labelling and relabelling procedure which uses the *spanning-tree* concept.

At every iteration, once the restricted Primal has



(a) Original network with mixed input-output conditions



(b) Canonical form of network (a)

Figure 5.

been solved without reaching optimality, an improved dual solution is computed, which in turn leads to a new restricted Primal. As the inspirer of the algorithm remarks, its efficiency is partly due to the fact that most of the network-type problems are loosely constrained, so that cost consideration (duality concepts) while trying to establish a feasible network-flow, is the dominant factor in obtaining optimality.

4.4 A Dual Algorithm

The relation of the primal and dual problem formulations, presented in (1) and (2) respectively, is given by the *Duality theorem* which states that *if there exists a finite optimum for one of the problems (1) and (2), then there exists a finite optimum for the other, and these two optima are equal; if one of them is found to have an unbounded objective, then the other is unfeasible.* Moreover, if **X** and **(V,U)** are the primal and dual solutions (in case of bounded problems), the *weak theorem of Complementary Slackness* asks that:

$$V_i - K_{ij}V_j < C_{ij} (U_{ij} = 0) \Rightarrow X_{ij} = 0 \qquad (a)$$

$$0 < X_{ij} < M_{ij} \Rightarrow V_i - K_{ij}V_j = C_{ij} \text{ and } U_{ij} = 0 \qquad (b)\,(5)$$

$$U_{ij} > 0\,(V_i - K_{ij}V_j = C_{ij} + U_{ij}) \Rightarrow X_{ij} = M_{ij} \qquad (c)$$

A **basic dual feasible solution** is a set of V_i, U_{ij} for which there are exactly $|V|$ (V's cardinal number) branches satisfying the relation:

$$V_i - K_{ij}V_j = C_{ij},\ U_{ij} = 0 \, (6)$$

Given a basic dual feasible solution, the set of real X_{ij} satisfying (5a,c) and the continuity principle (1b) is called the **network pseudoflow** related to that basic dual feasible solution. Furthermore, **branch coflow** δ_{ij} is defined to be the quantity:

$$\delta_{ij} = V_i - K_{ij}V_j - C_{ij} \, (7)$$

The data structures that are used in the algorithm system form the basis topology theorem and network concepts. Data that describe the network topology are static, while those which concern the basis representation are dynamically reorganized at each iteration. The algorithm begins with a dual feasible solution which is assumed to be available; it has already been pointed out that the method is efficient when applied to post-optimization cases. (If this is not the case, an initial basic dual feasible solution is obtained by introducing artificial loop-branches at every node, with very high cost and non-unary gain. V_i's are selected such that loop-branch coflows are priced at zero. If the problem is feasible, their high cost will keep these branches out of the optimal basis). Once an initial basic dual feasible solution is available, pseudoflows of the network branches are calculated by making use of the complementary slackness.

Specifically, branches with positive coflow are assigned a pseudoflow equal to their capacity, while those with negative coflow are set to zero. The pseudoflows of the remaining basic branches are calculated such that they satisfy the continuity principle. The basis topology allows for an efficient organization of the computations involved: in each basic connectivity component, the calculation of the branch pseudoflows of every basic subtree hanging from a cyclic node can be performed starting from its end-nodes and progressing to the root, the same principle at all loop-nodes of the component leads to a well-defined system of equations, is unknown to the loop-branch pseudoflows. This procedure, combined with the non-degeneracy of the basic loops, implies that for a given basic dual feasible solution, the set of related pseudoflows is single-valued. If the resulting basic flows are found to be within their boundaries, the Duality Theorem guarantees the optimality of the solution, and the algorithm terminates. Otherwise, a new basis has to be produced.

The branch (i_0, j_0) to leave the basis is the one that violates most the capacity constraints. The determination of the branch to enter the basis relies on a mechanism that ensures dual feasibility of the resulting basis. Initially, the non-basic branch coflow variations to a hypothetical *unary* variation of the coflow of the outgoing branch are calculated, with the rest of the basic branch-coflows remaining at zero. Taking into consideration the relation (7) $\Delta\delta_{ij} = \Delta V_i - K_{ij}\,\Delta V_j$ and the calculation previously mentioned, the problem reduces to the solution of the system of equations:

$$\Delta V_i - K_{ij}\Delta V_j = 0 \quad (i,j) \in B_b - \{(i_0, j_0)\}$$

$$\Delta V_{i(0)} - K_{i(0)j(0)}\Delta V_{j(0)} = 1 \quad (8)$$

in ΔV_i. It is a well-defined system, and the basic property of the branches involved implies the uniqueness of its solution. A pattern similar to that described for the computation of the pseudoflows can be used for a fast solution of this system. Afterwards, the maximal variation that sets one (or more in case of degeneracy) non-basic branch coflow to zero, leaving the others at their previous signs, is performed. The branch whose coflow is set to zero, enters the basis. Dual variables are assigned their new values, and data structures storing basis information are appropriately updated.

The efficiency and promptness of the algorithm results from the exploitation of the fact that only the connectivity components where the outgoing and entering branches belong to, are actively involved in the required computations and basis updating. At that point, a new iteration begins. It can be shown that the dual objective function increases at every iteration of the algorithm (or it remains constant, in case of degeneracy). So, convergence to optimality is achieved in a finite number of iterations. On finishing the presentation of the dual method, it is worth-mentioning that by making use of the same concepts and data structures, the *Primal* Simplex method has been tailored, in a similar fashion, to an efficient solution of the GN optimal flow problem.

4.5 The Relaxation Method

This approach makes use of the non linear *dual function* of the initial problem [20]. This is introduced by the following definitions:

First, we define the **Lagrangian function** of problem (1) as:

$$\mathfrak{L}(\mathbf{X},\mathbf{P}) = \sum_{(i,j)\in B} C_{ij}X_{ij} + \sum_{i\in N} P_i \left(\sum_{(m,i)\in B} K_{mi}X_{mi} - \sum_{(i,m)\in B} X_{im}\right) =$$

$$= \sum_{(i,j)\in B} (C_{ij} + K_{ij}P_j - P_i)X_{ij} \quad (9)$$

where **P** is the vector of **Lagrange multipliers** of constraints (1b), also called the **prices** of the nodes. Then, the **dual functional q** is given by

$$q(\mathbf{P}) = \min_{L_{ij}\le X_{ij}\le M_{ij}} \mathfrak{L}(\mathbf{X},\mathbf{P}) = \sum_{(i,j)\in B} q_{ij}(P_i - K_{ij}P_j)$$

(10)

where

$$q_{ij}(P_i - K_{ij}P_i) = \min_{L_{ij}\le X_{ij}\le M_{ij}} \left\{(C_{ij} + K_{ij}P_j - P_i)X_{ij}\right\} = \begin{cases}(C_{ij} - t_{ij})M_{ij} \text{ if } t_{ij} \ge C_{ij} \\ (C_{ij} - t_{ij})L_{ij} \text{ if } t_{ij} \le C_{ij}\end{cases}$$

(11)

and

$$t_{ij} = P_i - K_{ij}P_j \qquad (12)$$

for all $(i,j) \in B$, is the so-called **tension vector** corresponding to **P**.

In order that the above definitions should be valid, the right-hand-side vector of equation (1b) must be equal to zero. This can be achieved by a procedure similar to the canonical transform of Jewell's algorithm, appropriately exploiting the lower bounds L_{ij} and the fact that a single branch-loop with a non-unary gain operates as a source or a drain for the entire network. Then, we define the **deficit** of node i associated with a certain flow pattern as:

$$d_i = \sum_{(i,m)\in B} X_{im} - \sum_{(m,i)\in B} K_{mi}X_{mi} \quad (13)$$

From the above definitions it is obvious that q can be considered as a function in **t**, as well. The principal notion of the algorithm is that of the **ascent direction** of the dual functional, associated with a connected strict subset S of V, expressed by the vector $\vartheta = \{\vartheta_{ij} / (i,j) \in B\}$, where:

$$\vartheta_{ij} = \begin{cases} K_{ij}u_j & \text{if } i \notin S, j \in S \\ -u_i & \text{if } i \in S, j \notin S \\ K_{ij}u_j - u_i & \text{if } i \in S, j \in S \\ 0 & \text{otherwise} \end{cases} \quad (14)$$

In relation (14), u_i's are a set of positive numbers which are calculated for all nodes of subset S, with the aid of a spanning tree T of it, such that:

$$u_i - K_{ij}u_j = 0 \quad \text{for all } (i,j) \in T \qquad (15)$$

The **directional derivative** of the dual function at point **t**, in direction ϑ , is:

$$C(\vartheta,t) = \sum_{(i,j)\in B} \lim_{\alpha \to 0^+} \left[q_{ij}\left(t_{ij} + \alpha\,\vartheta_{ij}\right) - q_{ij}\left(t_{ij}\right)\right] / \alpha =$$

$$= \sum_{i \in S} u_i d_i -$$

$$- \sum_{\substack{i\in S, j\notin S \\ (i,j):\text{active}}} (X_{ij} - L_{ij})u_i$$

$$- \sum_{\substack{i\notin S, j\in S \\ (i,j):\text{active}}} (M_{ij} - X_{ij})K_{ij}u_j \qquad (16)$$

$$- \sum_{\substack{i\in S, j\in S \\ (i,j):\text{active}, u_i > K_{ij}u_j}} (X_{ij} - L_{ij})(u_i - K_{ij}u_j)$$

$$- \sum_{\substack{i\in S, j\in S \\ (i,j):\text{active}, u_i < K_{ij}u_j}} (M_{ij} - X_{ij})(K_{ij}u_j - u_i)$$

The *weak duality of Lagrangian Relaxatio* postulates that at optimality

$q(P_0)=\max_P q(P)=c$ (17)

The algorithm initiates with a set of dual prices **P** and flows **X** satisfying complementary slackness, but not the constraints (1b). At every iteration the algorithm proceeds towards optimality by either a *redistribution of flow* which reduces nodal deficits, or a *revision of the nodal prices*, so that the dual functional increases, whichever comes first. Specifically, at every iteration a non-balanced node is spotted, and in continuation, a scanning mechanism is put to work in order to determine one of the above-mentioned possibilities of improvement. A deficit reduction can be achieved any time a route to a node with deficit of opposite sign to that of the set-out node, is established or an appropriate cyclic structure is developed; a dual functional increase is possible if the directional derivative associated with the scanned subset of nodes S is positive. It should be mentioned that primal-dual algorithms operate in a similar way but the direction of correction is that of the maximal rate of ascent. However, according to the algorithm's inspirers, its efficiency stems from the fact that in many iterations the direction subset S may consist of the initial node only, which is attributed by the characterization *coordinate ascent algorithm*. In that case , it is shown that the negative of the nodal deficit, $-d_s$, is a *subgradient* of the dual functional at $\mathbf{P_S}$ in the sth coordinate direction. A final remark concerning the algorithm's convergence, is that that the algorithm is proven to be finite if ε *-optimal* solutions are searched, with ε defined arbitrary small. At every iteration, vectors **P**, **X** must satisfy a modified version of complementary slackness, the so called ε *complementary slackness*, defined by

ε -inactive if $t_{ij} < C_{ij} - \varepsilon$
ε -active if $C_{ij} - \varepsilon \leq t_{ij} \leq C_{ij} + \varepsilon$ (18)
ε -hyperactive if $C_{ij} + \varepsilon < t_{ij}$

Then, the produced solution will be within a radius of $\varepsilon \sum_{(i,j)\in B} (M_{ij} - L_{ij})$ from the optimal one.

RELAX-II, an implementation code of the algorithm, is reported to be an order of magnitude faster than the latest state-of-the-art codes, which implement Primal and Primal-Dual methods.

5. Applications and Conclusions

Generalized networks can be used to model numerous problems to which there are no pure network equivalents. Essentially there are two ways in which GN multipliers can function: they can simply act to modify the amount of flow transversing the branch, representing for example phenomena of evaporation, seepage, deterioration, breeding, etc., or they can transform flow from one type into another, thus modelling the processes of manufacturing, production, conversion of fuel to energy, etc. The range of applications of GNs in various areas of Operations Research (O.R.) is the following:

Scheduling and Planning:

Scheduling and Planning models usually require several time periods for their basic structure. Observing that they possess a network structure, one can see that the GN model has a significant impact on their solution. For one thing, large-scale network applications can be solved easily, and the level of detail provided to the model can be expanded considerably. Secondly, planning tools with quite fast response time may be designed, so that they are executed interactively. Outstanding areas of this category where considerable work has been done are *Hydroelectric Scheduling* and *Air Traffic Control.*

Finance:

Mathematical planning models are an established part of financial analysis. Work in this area has begun since 1950. Several authors have proposed network models in which branch multipliers play a primary role. Some suggest that multipliers can be used for translating currencies across countries (e.g. dollars to Deutsch Marks); others have designed multiperiod models in which interests, dividends and loans are modelled by means of multipliers. *Cash flow management* provides an ideal application for this methodology.

Equilibrium Models:

There are many instances of such models; some of them are summarized below.

- *Traffic equilibrium*, in which case branches represent transportation paths (e.g. highways) and nodes are the connections between paths.
- *Market equilibrium*, with nodes indicating spatially separated markets for products, and branches indicating their interdependence .
- *Demographic equilibrium*, with nodes indicating distinct demographic zones, and branches indicating possible migration patterns among the zones.
- *Water/gas pipeline distribution systems*, with branches representing the pipes, and nodes indicating the connection points.

Statistics and Large Databases:

The timely collection of very large databases has become increasingly important over the past two decades. Many government agencies and private companies routinely depend upon these files for maintaining their operations. One important class of databases is known as *microdata* whereby the file consists of a large number of individual decision units-individuals, families, corporations, etc. Typically, microdata files range in size from one thousand to over one million observations. There are numerous steps involved, several of which employ network optimization. A representative example of this category is the *estimation of Social Accounting Matrices and File Merging* (SAM).

Distribution and Logistics

Models of integrated production, inventory and distribution are often based on a network form. The flow of materials within the factory, and the transport of the final product through distribution centers onto the market, present an appealing case where a large part of the problem can be depicted as a graph. The commensurate gain in efficiency lets the modeller room for including and modelling such features as more time periods, desegregate customer zones, additional potential sites for warehouses, or even a more realistic cost structure.

Of course, there are more active areas of research in Engineering Design and O.R., in which GNs play an important role for problem solvability. An extended version of the GN's, the **0-1 GN problem**, broadens the scope of the problems to be treated so that classical *Integer Programming* (IP) and *Mixed IP* (MIP) problems are included. According to this formulation, certain branch-flows are obliged to take the values of only 1 and 0, and the incorporation into the network of structures similar to those of Figure 6, can lead to the integrality of all the variables, with only a small number of them explicitly stated as being of integer type. In case that a *Branch and Bound* method is used for the solution of the problem, this fact results in a considerable reduction of the underlying *enumeration-tree* structure and thus, in very fast solution times.

Taking the integer variables to be decision variables, a new group of problems can be faced as 0-1 GN's, including job assignment to computers, the design and operation of capacitated communication networks, financial capital allocation, plant location, energy models and physical distribution.

GENETEXP has been used by the authors in transportation planning problems of large size with real data and nodes provided by an European supply-distribution chain. The results are quite satisfactory from both the time-accuracy and the conflict resolution points of view. Of course work is still in progress for further enhancing the tool (especially its KB component) and for improving the user-interface module.

Acknowledgment
Part of the work described in this paper was supported by ESPRITCIM Project 2277 (CMSO).

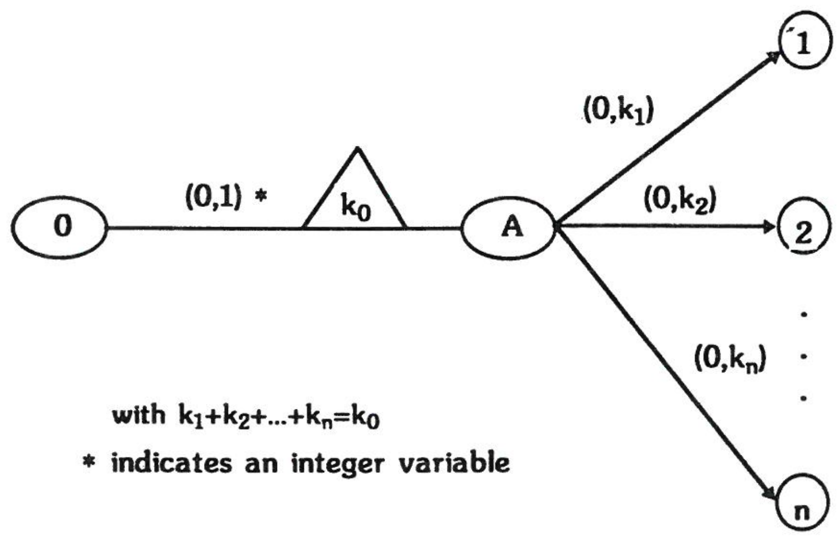


Figure 6. 0-1 GN Integrality Structure

REFERENCES

1. O'KEEFE, R.M. , **Expert Systems and Operational Research- Mutual Benefits**, J. OPER. RES. SOC. , Vol. 36, 1995, pp. 125-129.
2. SIMON, H.A. , **Two Heads Are Better Than One: The Collaboration Between AI and OR**, INTERFACES , Vol. 17, 1987, pp. 8-15.
3. WHITE, C.C. and SYKES, E.A. , **A User Preference Guided Approach to Conflict Resolution in Rule-Based Expert Systems**, IEEE TRANS. SYST. MAN CYBERN , Vol. SMC-16, 1986, pp. 276-278.
4. LEHNER, P.E. , PROBUS, M.A. and DONNEL, M.L. , **Building Decision Aids: Exploiting the Synergy Between Decision Analysis and Artificial Intelligence**, IEEE TRANS. SYST. MAN CYBERN. , Vol. SMC-15, No. 4, 1985, pp. 469-474.

5. WEINROTH, J. , **Model-Based Decision Support and User Modifiability**, IEEE TRANS. SYST. MAN CYBERN. , Vol. 20, No. 1, 1990, pp. 513-518.
6. MOSER, J.C. , **Integration of AI and Simulation in A Comprehensive Decision Support System**, SIMULATION , Vol. 47, 1986, pp. 223-239.
7. ELAM, J. , KLINGMAN, D. and SCHNEIDER, R. , **Decision Support Systems for Tactical Logistics Planning**, Proc. Natl. Council for Physical Distribution , San Francisco, CA, USA, October 1982.
8. DE, S., NOF, S.Y. and WHINSTON, A.B. , **Decision Support in Computer-Integrated Manufacturing**, DECISION SUPPORT SYSTEMS , Vol. 1, 1985, pp. 37-56.
9. TZAFESTAS, S.G. and HATZIVASSILIOU,F., **Human-Computer Interface: Artificial Intelligence and Software Psychology Issues**, Proc. Europ. Conf. on Geographical Information Systems , (EGIS` 90), Amsterdam, April, 1990.
10. HATZIVASSILIOU, F. and TZAFESTAS, S.G. , **Human-Computer Intelligence Interface. Design Using Software Psychology**, Proc. 5th Intl. Symp. on Computer and Information Sciences (ISCISV), Cappadocia, Turkey, October 1990.
11. RICH, E. , **Natural Language Interface**, IEEE COMPUTER , September 1984, pp. 39-47.
12. LIGEZA, A. , **Expert Systems Approach To Decision Support**, EUROP. J. OPER. RES. , Vol. 37, 1988, pp 100-110.
13. TZAFESTAS, S.G. and LIGEZA, A. , **Expert Control Through Decision Making**, IMACS/IFORS Int. Symp. on AI, Expert Systems and Languages in Modeling and Simulation , Barcelona, 1987.
14. TZAFESTAS, S.G. and LIGEZA, A. , **A Framework for Knowledge-Based Control**, J. INTELL. AND ROBOTIC SYST. , Vol. 1, 1989, pp. 407-425.
15. GLOVER, F. , HULTZ, J. and KLINGMAN, D. , **Improved Computer-Based Planning Techniques**, Part I, INTERFACES , Vol.8, No.4, 1978, pp. 16-25 .
16. STUTZ, J. , **Generalized Networks: A Fundamental Computer-Based Planning Tool**, MANAGEMENT SCI. , Vol. 23, No. 12, 1978, pp. 1209-1220.
17. JENSEN, A.P.and BHAUMIK, G. , **A Flow Augmentation Approach to the Network with Gains Minimum Cost Flow Problem**, MANAGEMENT SCI., Vol. 23, No. 6, 1977, pp. 631-643.
18. JEWELL, S.W. , **Optimal Flow Through Networks with Gains**, OPERATIONS RES. , Vol. 10, 1962, pp. 476-499.
19. TZAFESTAS, S.G. , KAPSIOTIS, G. and REVELIOTIS, S. , **A Dual Algorithm for Post-Optimization of the Generalized Network Optimal Flow Problem**, OPERATIONS RESEARCH 1990: DGOR-GMÖOR-OGOR-SVOR Intl. Symp. on O.R. , Vienna, Austria, August 1990.
20. PERTSEKAS, P.D. and TSENG, P. , **Relaxation Methods for Minimum Cost Ordinary and Generalized Network Flow Problems**, OPER. RES., Vol. 36, No. 1, 1988, pp. 93-114.
21. TZAFESTAS, S.G. , REVELIOTIS, S. and KAPSIOTIS, G., **GENET OPTIMIZER: A Tool for Optimized Transportation Planning**, NTVA Report, Comp. Sci. Div. , Athens, Greece, August 1990.
22. TZAFESTAS, S.G., ELIANOS, A. and SAROGLOU, G. , **BIOEXP: An Expert System for Therapeutical Use in Pneumonia Treatment**, Proc. IMACS/IFAC Symp. on Mathem. and Intell. Models in Syst. Simul. , Brussels, Belgium, September 1990.
23. TZAFESTAS, S.G. and KOSTANTINIDIS, N., **ENGEXP: An Expert System for Equipment and Automotive Engine Fault Detection and Repair**, NTVA Report , Comp. Sci. Div. , Athens, Greece, September 1990.